BIOINFORMATICS

Vol. 00 no. 00 2010
Pages 1–9

Markov Dynamic Models for Long-Timescale Protein Motion

Tsung-Han Chiang [1]*, David Hsu [1] and Jean-Claude Latombe [2]

[1]Department of Computer Science, National University of Singapore, Singapore 117417, Singapore
[2]Department of Computer Science, Stanford University, Stanford, CA 94305, USA

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Molecular dynamics simulation is a well-established method for studying protein motion at the atomic scale. However, it is computationally intensive and generates massive amounts of data, the sheer size of which often becomes an obstacle to biological insights. One way of addressing the dual challenges of computation efficiency and data analysis is to construct simplified models of protein motion at long timescales, as many important kinetic and dynamic properties of proteins ultimately depend on such motions. In this direction, we propose to use Markov models with hidden states, in which the Markovian states represent potentially overlapping probabilistic distributions over a protein's conformation space. We also propose to evaluate the quality of a model by its ability to predict long-timescale protein motions. Our method was tested on 2-D synthetic energy landscapes and two extensively studied proteins, alanine dipeptide and villin. One interesting finding is that although a widely accepted model of alanine dipeptide contains 6 states, a simpler model with only 3 states is equally good for predicting the long-timescale motions. This finding highlights the importance of a principled criterion for evaluating model quality. We also used the constructed Markov models to estimate important kinetic and dynamic quantities for protein folding, in particular, mean first-passage time. The results are consistent with available experimental measurements.

Contact: chiangts@comp.nus.edu.sg

1 INTRODUCTION

Protein motion is the aggregate result of complex interactions among individual atoms of a protein at timescales ranging over several orders of magnitudes. Thermal fluctuations, which occur in picoseconds (10^{-12} seconds), are small-amplitude, uncorrelated, harmonic motions of atoms, but they eventually provide the protein with enough momentum to overcome energy barriers between meta-stable states. In contrast, biologically significant conformational motions, which occur in microseconds to milliseconds, are often large-scale, correlated, anharmonic motions between meta-stable states. For example, in a folded protein, they may occur between binding and non-binding states. The wide range of timescales and complex relationships among the motions at different timescales make it difficult to capture the biologically significant, long-timescale dynamics of protein motion in a compact and efficient model.

Molecular dynamics (MD) simulation is a well-established method for studying macromolecular motion at the atomic scale [23]. However, it requires a detailed energy function, and the equations of motion must be integrated with a time step much shorter than the timescale of atomic thermal fluctuations. Today's computers can generate roughly a few nanoseconds of simulation trajectories in a day, which is insufficient for capturing events of biological significance. Distributed computing [13] and specialized computer architectures [12] speed up MD simulation significantly, but the sheer size of data generated is a major hurdle that prevents biological insights. One way of addressing both the issues of computational efficiency and data analysis is to construct a simplified model that captures the essential features of protein motion at long timescales. Markov dynamic models (MDMs) provide a promising direction towards this goal.

An MDM of a system—here, a protein—can be represented as a directed graph. Each node of the graph represents a state s of the system, and each edge represents a transition from state s to s'. Each edge $(s,\ s')$ is also assigned the probability that the system transitions from s to s' in one time step. MDMs have several advantages for modeling protein motion. First, they are probabilistic and thus naturally capture the stochasticity of protein motion. Second, MDMs represent states explicitly. This makes them potentially easier to understand and faster to simulate. Finally, there are standard algorithmic tools, *e.g.*, first-step analysis [26], for exploiting MDMs without expensive explicit simulation.

A key issue in constructing an MDM is the choice of states. What are the Markovian states of a protein needed to accurately model its long-timescale dynamics? One contribution of this work is to have states represent not individual protein conformations [3, 25], not even disjoint regions of the conformation space [7], but overlapping probabilistic distributions of conformations. This choice reflects the view that a conformation does not contain enough information to be uniquely assigned to a *single* state. Although this may seem odd at first, it is in fact quite natural in modeling many physical systems. For example, suppose that we want to classify some physical objects into two states, table or chair. For a cubic object one meter in size, if we see a meal on top of it, we may consider it a table; if we see someone seated on it, we consider it a chair. So, a cube in itself cannot be assigned a single state because there is insufficient

[*]to whom correspondence should be addressed

information. In many cases, acquiring and representing missing information, if at all possible, is much more costly than capturing it in a probabilistic distribution. Hence our choice of Markovian states that represent probabilistic distributions over the protein conformation space. This choice leads to MDMs with *hidden states*, formally, hidden Markov models (HMMs). In this paper, we present a method to automatically construct an HMM of the long-timescale dynamics of a protein from a dataset of MD simulation trajectories.

Another key question is how to measure the quality of a model. A good model enables us to predict biologically relevant quantities of protein motion accurately and efficiently. However, a particular model may do well for one quantity, but poorly for another. Also, we may not know in advance the quantities to be predicted when constructing a model. Another contribution of this work is to evaluate the quality of a model by its ability to predict long-timescale protein motions, as many interesting kinetic and dynamic properties of proteins ultimately depend on such motions. Specifically, we score an HMM probabilistically by its likelihood for a test dataset of MD trajectories. Using this criterion, we are able to select models that make good predictions on ensemble quantities characterizing the folding of alanine dipeptide and villin, two extensively studied proteins.

We also present an efficient algorithm for computing mean first-passage time from any conformation of a protein to the folded conformation, using an HMM of protein dynamics.

In the following, Section 2 reviews previous work. Section 3 describes our model of long-timescale protein motion. Section 4 presents an efficient algorithm for computing ensemble properties of protein folding, in particular, mean first-passage time. Section 5 describes the algorithm for model construction, Section 6 presents the results on synthetic landscapes, alanine dipeptide and villin. Section 7 points out future research directions.

2 RELATED WORK

2.1 Graphical Models of Protein Motion

Our work proceeds from a series of developments that started with adapting probabilistic roadmap (PRM) planning [16] from robotics to model molecular motion. PRM is a class of algorithms for controlling the motion of complex robots.

Roadmap models. A *probabilistic roadmap* for a robot is an undirected graph. Each node q of the graph represents a valid robot configuration sampled randomly from the space of all valid robot configurations, and each edge between two nodes q and q' represents a valid motion between the conformations corresponding to q and q'. PRM planning is currently the most successful approach for motion planning of complex robots with many degrees of freedom. The PRM approach was adapted to model and analyze the motion of a flexible ligand binding with a protein [24]. The modified roadmap is a directed graph, in which each node represents a sampled ligand conformation and each directed edge represents the transition from one ligand conformation to another. Each edge is also assigned a *heuristic weight* measuring the "energetic difficulty" of the transition. This approach was used to predict active binding sites of a protein [24] and the dominant order of secondary structure formation in protein folding [2].

From roadmaps to MDMs. To capture the stochasticity of molecular motion, a roadmap model was transformed into an MDM by treating each roadmap node as a state and assigning each edge (q, q') the *transition probability* derived from the energetic difference between the conformations q and q' [3]. We call this model a *point-based MDM*, as each state represents a single conformation. First-step analysis was applied to the MDM to compute efficiently the p-fold value, a theoretical measure of folding progress [3]. This approach was later improved to predict experimental measures of folding kinetics and dynamics, such as folding rates and ϕ-values [6]. An improved sampling method generates the states of an MDM using MD simulation data [25]. It obtains better coverage of the biologically relevant part of the protein conformation space, and also captures the protein dynamics more accurately by labeling each edge (q, q') of the MDM with the average transition time between the conformations q and q'.

From point-based to cell-based MDMs. In an point-based MDM, a state represents a single protein conformation. However, a conformation rarely contains enough information to guarantee the Markovian property, a fundamental MDM assumption requiring that the future state of a protein depends on its current state only and not on the past history. Consequently a large number of states are needed to construct a good MDM. This drawback led to *cell-based* MDMs [7], in which each node represents a *region* (a cell) of the conformation space. A cell s roughly matches a basin in a protein's energy landscape. The protein interconverts rapidly among different conformations within a basin before it eventually transitions to another basin. The assumption is that after many interconversions within s, the protein "forgets" the history of how it entered s and transitions into s' with probability depending on s only. A cell-based MDM can be built from a set of MD simulation trajectories. To satisfy the Markovian property well, conformations along these trajectories are grouped into clusters in such a way that maximizes self-transition probabilities for the states in the MDM [7]. More recent work extended this approach to build MDMs at multiple resolutions through hierarchical clustering [15].

2.2 Other Approaches

Many other approaches have been explored to model and understand protein motion. See [10] for a recent survey. Here, we only mention a few that are more closely related to our work.

Normal mode analysis [19] and related approaches, such as elastic network models [14], simplify the complex dynamic law that governs protein motion by approximating it near an equilibrium conformation. One advantage is that they capture the geometry and mass distribution of a protein structure compactly in a relatively simple model. However, they are relatively accurate only in the neighborhood of the equilibrium conformation.

Another approach for building simple dynamic models is to find reaction coordinates [20]. Significant events are described along a carefully chosen one-dimensional reaction coordinate. The choice of this coordinate, however, requires *a priori* understanding of the protein motion. Furthermore, not all proteins can have their motions described and understood along a single coordinate.

Instead of building simplified dynamic models, one may analyze MD simulation data directly through dimensionality reduction methods [1, 27]. Unlike normal mode analysis, this approach provides a global view of protein motion. It may also help identify a good reaction coordinate. However, this approach does not provide a predictive model that generalizes the simulation data. Nor does it identify interesting states of protein dynamics.

3 MARKOV DYNAMIC MODELS WITH HIDDEN STATES

A MDM Θ of a protein can be represented as a weighted directed graph. A node s of Θ represents a state of the protein, and a directed edge (s, s') from node s to s' represents a transition between the corresponding states. Each edge (s, s') is assigned a weight $a_{ss'}$ representing the probability that the protein in state s transitions to state s' in a time step of fixed duration h. The probabilities associated with the outgoing edges from any node s must sum up to 1. The duration h is the *time resolution* of the model.

An MDM describes how the state of the protein changes stochastically over time. Given an initial state s_0 of the protein at time 0, we can use the MDM to predict a sequence of future states $s_1, s_2, \ldots$, where s_t is the state of the protein at time $t \times h$, where $t = 1, 2, \ldots$. If $s_t = s$, then we predict the next state s_{t+1} by choosing an outgoing edge (s, s') from s with probability $a_{ss'}$ and setting $s_{t+1} = s'$. The simple and explicit structure of MDMs allows such predictions to be computed efficiently.

In a point-based MDM, a state represents a single conformation. In a cell-based MDM, a state represents a set of conformations (see Section 2.1). The definition of states is critical. The choice of a single conformation as a state is more precise and informative than the choice of a set of conformations. However, it often leads to violation of the Markovian property and consequently reduces the predictive power of the MDM. We now address the delicate question of defining the states.

3.1 Why Hidden States?

By defining states as regions of the protein conformation space, rather than single conformations, cell-based MDMs achieve the dual objectives of better satisfying the Markovian assumption and reducing the number of states. This is a major step forward. However, cell-based MDMs still violate the Markovian assumption in a subtle way. Consider a protein at a conformation q near the boundary of a cell. The future state of the protein depends not only on q, but also on the protein's velocity, in other words, on the past history of how the protein reaches q. By requiring each conformation to belong to a *single* state, cell-based MDMs violate the Markovian assumption, especially near the cell boundaries. Similar violations also occur in cells corresponding to shallow energy basins, where the protein's energy landscape is flat.

One way of fixing such violations is to define more refined states using information on both conformation and conformational velocity. However, this necessarily increases the number of states, thus partially reversing a key advantage of cell-based MDMs. Furthermore, a much larger dataset is needed for model construction in order to capture the detailed transition probabilities among the refined states. In contrast, we propose to assign a conformation to *multiple* states and use probabilities to capture the uncertainty of state assignment. This leads to an MDM with hidden states, formally, an HMM. Our HMM for protein dynamics is specified as a tuple $\Theta = (S, C, \Pi, A, E)$:

- The set of states $S = \{s_i \mid i = 1, 2, \ldots, K\}$;
- The conformation space C of a protein;
- $\Pi = \{\pi_i \mid i = 1, 2, \ldots, K\}$, where π_i is the prior probability that the protein is in state $s_i \in S$ at time $t = 0$;
- $A = \{a_{ij} \mid i, j = 1, 2, \ldots, K\}$, where $a_{ij} = p(s_j|s_i)$ is the probability of transitioning from state $s_i \in S$ to $s_j \in S$ in a single time step of duration h;
- $E = \{e_i \mid i = 1, 2, \ldots, K\}$, where $e_i(q) = p(q|s_i)$ is the *emission probability* of observing conformation $q \in C$ when the protein is in state $s_i \in S$.

The state space S is discrete, while the conformation space C is continuous. Intuitively each state $s_i \in S$ loosely matches an energy basin of the protein, and the corresponding emission probability $e_i(q) = p(q|s_i)$ connects states with conformations by modeling the distribution of protein conformations within the basin.

In an HMM, we cannot assign a unique state for a given conformation q. Instead we calculate $p(s_i|q)$, the probability that q belongs to a state s_i. The uncertainty in state assignment arises because at a conformation q, the protein may have different velocities, as well as other differences, which we choose not to model or do not know about. We model the uncertainty due to this lack of information with the emission probability distributions.

In contrast, a cell-based MDM partitions C into disjoint regions $C_1, C_2, \ldots$, and each state s_i represents a region C_i. So we can assign a conformation q to a unique state. If we define e_i as a step function such that $e_i(q)$ is a strictly positive constant for $q \in C_i$ and 0 otherwise, then the states are no longer hidden, and our model degenerates into a cell-based MDM. Our distribution-based models are therefore more general than cell-based MDMs.

3.2 What is a good model?

Another difficulty with cell-based MDMs is the lack of a principled criterion for evaluating model quality. To satisfy the Markovian property well, the algorithm for constructing cell-based MDMs maximizes the self-transition probabilities for the states in the model [7]. This criterion, however, results in the paradoxical conclusion that a trivial one-state model is perfect, as all transitions are self-transitions. Since simple models are usually preferred, how do we determine that a simple model is as good as or better than a more complex one?

Our goal is to build a model Θ of the long-timescale dynamics of a protein from a given dataset D of MD simulation trajectories. The model Θ is then used to predict various kinetic and dynamic properties of protein motion, such as mean first-passage times [18], p-fold values [9], transition state ensembles [18], *etc.*. A model Θ_1 has stronger predictive power than a model Θ_2, if Θ_1 predicts the kinetic and dynamic properties more accurately than Θ_2. Clearly it is impossible to check the predictive power of a model Θ on all such properties, as we may not even know them all in advance. However, since many kinetic and dynamic properties are determined by protein motion trajectories, we can check instead the ability of Θ to predict these trajectories. In our HMM framework, we do this by calculating the likelihood $p(D|\Theta)$, which is the probability that a dataset D of MD simulation trajectories occur under the model Θ. The likelihood $p(D|\Theta)$ measures the quality of Θ.

Specifically, let $D = \{D_i \mid i = 1, 2, \ldots\}$ be a dataset of trajectories. Each trajectory D_i is a sequence of protein conformations $(q_0, q_1, \ldots, q_T)$, where q_t is the protein conformation at time $t \times h$. The likelihood of Θ for D_i is

$$p(D_i|\Theta) = \sum_{Q \in S^T} \Big(p(s_0) \prod_{t=1}^{T} p(s_t|s_{t-1}) \prod_{t=0}^{T} p(q_t|s_t) \Big), \quad (1)$$

where s_t is the state of the protein at time $t \times h$ and $p(s_0)$, $p(s_t|s_{t-1})$, and $p(q_t|s_t)$ are given by the model parameters Π, A, and E of Θ, respectively [4]. The summation $\sum_Q$ is performed over all possible state assignments $Q = (s_0, s_1, \ldots, s_T) \in S^T$ to the conformations $(q_0, q_1, \ldots, q_T)$ in D_i. The likelihood of Θ for the entire dataset D is simply $p(D|\Theta) = \prod_i p(D_i|\Theta)$.

In contrast to the cell-based MDM, the likelihood $p(D|\Theta)$ provides a quantitative measure of model quality and enables us to compare models with different number of states. This is possible, because our model uses emission probabilities $e_i(q) = p(q|s_i)$ to connect states with conformations, while a cell-based MDM does not. The likelihood criterion shows that a single-state MDM is in fact not good. Although the transition probabilities $p(s_t|s_{t-1}) = 1$ for all t, the emission probabilities $p(q_t|s_t)$ are small, because the model relies on a single state to capture variability over the entire conformation space. Hence the overall likelihood $p(D|\Theta)$ is small.

4 MODEL EXPLOITATION

Before discussing how to automatically construct our model from MD simulation data, let us first consider how to make use of it.

First, our MDM is a graphical model. We can gain various insights of the underlying folding process by inspecting the structure and the edge weights of the graph. We give an example in Section 6.

Next, our MDM is generative and can be used for simulation. To generate a simulation trajectory of length T, we first sample a state sequence $(s_0, s_1, \ldots, s_T)$ from the model. We sample the initial state s_0 according to a prior distribution adapted to the environmental condition of the biological events under study. We then sample each subsequent state s_t conditioned on the previous state s_{t-1} according to the transition probabilities A. After obtaining the state sequence, we generate the trajectory $(q_0, q_1, \ldots, q_T)$ by sampling each q_t conditioned on s_t with probability $p(q_t|s_t)$ according to the emission probabilities E.

Furthermore, an important advantage of MDMs is that they can be analyzed systematically without explicitly generating simulation trajectories. In particular, our model allows for efficient computation of *ensemble properties* of protein folding. Ensemble properties, such as mean first-passage time (MFPT) and p-fold value, characterize the average behavior of a folding process over myriad pathways at the microscopic level. In principle, we can compute ensemble properties by simulating many individual pathways and then averaging over them, but explicit simulation is computationally expensive. In the following, we describe a more efficient algorithm to compute MFPT using our model. P-fold value and other ensemble properties can be computed similarly.

The MFPT of a conformation q is the expected time for a protein to reach a folded conformation, starting from q. It measures the speed of folding. A straightforward way of estimating the MFPT of q is to simulate many folding trajectories, each starting from q and terminating upon reaching a folded conformation. The estimated MFPT is then the average length of these trajectories. This approach typically requires a huge number of simulation trajectories to get a reliable estimate for a single conformation q.

Instead we apply first-step analysis [26] from Markov chain theory to our model. It implicitly simulates infinitely many trajectories, resulting in much faster and reliable computation of MFPTs. However, since our model is constructed from a dataset of MD simulation trajectories, an interesting question is “How can the model give more reliable MFPT estimates than the simulation trajectories themselves?” Intuitively, the answer is that the model generalizes the data under the Markovian property and thus contains a lot more trajectories than the dataset used for model construction. For example, a dataset contains two trajectories with state sequences (s_0, s_1, s_2) and (s'_0, s_1, s'_2). Using the Markovian property, the model assumes that two additional state sequences (s_0, s_1, s'_2) and (s'_0, s_1, s_2) are also valid. Thus the model in fact contains exponentially more trajectories than the dataset and can give more reliable MFPT estimates provided that the Markovian property holds and that all the trajectories can be processed efficiently.

Our computation proceeds in two stages. First, we compute the MFPTs for all the states in S. Let $C_{\mathrm{F}} \subset C$ be the subset of folded conformations of a protein. Let γ_i be the first-passage time (FPT) of a folding trajectory that starts in state s_i. First-step analysis considers what happens in the very first time step of the folding trajectory:

- If the initial conformation $q_0 \in C_{\mathrm{F}}$, then obviously $\gamma_i = 0$. This event happens with probability $e_i(C_{\mathrm{F}}) = \int_{C_{\mathrm{F}}} e_i(q)\, dq$.
- If $q_0 \notin C_{\mathrm{F}}$, then γ_i depends on the MFPT of the state that the trajectory reaches after a one-step transition. This event happens with probability $1 - e_i(C_{\mathrm{F}})$.

The MFPT for s_i is $\bar{\gamma}_i = \mathrm{E}(\gamma_i)$, where the expectation is taken over *all* trajectories that start in s_i and end in C_{F}. By conditioning on the events in the first time step, we obtain the following equation for $\bar{\gamma}_i$:

$$\bar{\gamma}_i = 0 \cdot e_i(C_{\mathrm{F}}) + \Big(1 + \sum_{s_j \in S} p(s_j|s_i)\bar{\gamma}_j\Big) \cdot \big(1 - e_i(C_{\mathrm{F}})\big). \quad (2)$$

The values for $p(s_j|s_i)$ are given by the transition probabilities in A. The only unknowns in (2) are the MFPTs $\bar{\gamma}_i$ for $i = 1, 2, \ldots, K$. Since there is one such equation for each $\bar{\gamma}_i$, we get a linear system of K equations with K unknowns, which can be solved efficiently using standard numerical methods. The algebraic process of solving the linear system implicitly enumerates all possible state sequences of the folding trajectories in an efficient way.

After obtaining the MFPTs for the *states*, we compute the MFPT for a given *conformation* q_0. Let γ denote the FPT of a folding trajectory that starts at q_0. Conditioning on the initial state s_0 at $t = 0$, we see that the MFPT of q_0 is given by:

$$\mathrm{E}(\gamma|q_0) = \sum_{s_0 \in S} \mathrm{E}(\gamma|q_0, s_0) p(s_0|q_0). \quad (3)$$

We calculate $p(s_0|q_0)$ using the Bayesian rule:

$$p(s_0|q_0) = \frac{p(q_0|s_0)p(s_0)}{\sum_{s_0 \in S} p(q_0|s_0)p(s_0)}, \quad (4)$$

where $p(s_0)$ and $p(q_0|s_0)$ can be obtained from the prior probabilities Π and the emission probabilities E of the model, respectively. Calculating $\mathrm{E}(\gamma|q_0, s_0)$ is more subtle. Suppose that the initial state s_0 is some particular state $s_i \in S$. It is tempting to think that $\mathrm{E}(\gamma|q_0, s_0) = \bar{\gamma}_i$. This is incorrect, because $\bar{\gamma}_i = \mathrm{E}(\gamma|s_0)$ and the additional information provided by q_0 may alter the expected value of γ. To calculate $\mathrm{E}(\gamma|q_0, s_0)$, we condition once more on the state s_1 at time $t = 1$:

$$\mathrm{E}(\gamma|q_0, s_0) = \sum_{s_1 \in S} \mathrm{E}(\gamma|q_0, s_0, s_1) p(s_1|q_0, s_0) \quad (5)$$

$$= \sum_{s_1 \in S} \big(1 + \mathrm{E}(\gamma|s_1)\big) p(s_1|s_0), \quad (6)$$

where the last line follows from the conditional independence properties of HMMs [4]. Now the values for $\mathrm{E}(\gamma|s_1)$ can be obtained from the MFPTs $\bar{\gamma}_i$ where $i = 1, 2, \ldots, K$, and the values for $p(s_1|s_0)$, from the transition probabilities A. Substituting (4) and (6) into (3) gives us the desired result.

In practice, when we compare with experimental measures, we are interested in the MFPT for a region C' of C rather than a single conformation $q_0 \in C$. To calculate $\mathrm{E}(\gamma|C')$, we need to modify (3), (4), and (6) slightly by integrating q_0 over C'.

5 MODEL CONSTRUCTION

Under the likelihood criterion, we want to construct a model Θ that maximizes $p(D|\Theta)$ for a given dataset D of MD simulation trajectories. Expectation maximization (EM) is a standard algorithm for such optimization problems. However, EM is computationally intensive. It may also get stuck in a local maximum and fail to find the model with maximum likelihood. To alleviate these difficulties, we proceed in three steps. First, we preprocess the input trajectories to remove the "noise", *i.e.*, motions at timescales much shorter than those of interest. Next, we use K-medoids clustering to build an initial model Θ_0. Since clustering is much faster than EM, we run the clustering algorithm multiple times and choose the best result as Θ_0. This reduces the chance of ending up with a bad local maximum. Finally, we initialize EM with Θ_0 and search for the model with maximum $p(D|\Theta)$. Since both K-medoids clustering and EM are well known algorithms (see, *e.g.*, [4]), we only describe the relevant details of these steps below.

Data preparation. The time resolution h of the model should be compatible with the timescale of biological events under study. If h is too large, the resulting model may miss the events under study. If h is too small, the model will try to capture fine details at uninteresting short timescales and become unnecessarily complex with reduced predictive power. In our tests, a relatively wide range of h values led to models with similar predictive power. We typically set h to be 1/100 to 1/10 the timescale of interest. We then apply standard signal processing techniques [21] to smooth and downsample each trajectory in D so that the time duration between any two successive conformations along a trajectory is exactly h.

Emission probability distributions. The emission probability e_i models the distribution of protein conformations in state s_i. We approximate e_i with a Gaussian distribution:

$$e_i(q) = N(q|\mu_i, \sigma_i^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\big(-\frac{1}{2\sigma^2} d^2(q, \mu_i)\big), \quad (7)$$

where $d(q, \mu_i)$ denotes a suitable distance measure between the conformations q and μ_i. Other approximating distributions are possible. There are two main considerations in choosing the distribution: it should match the shape of s_i well and be simple enough to be learned effectively with a limited amount of data.

Initialization. The states in our model roughly correspond to energy basins. Within a basin, a protein interconverts rapidly, which allows inter-state protein motions to satisfy the Markovian property well. Rapid interconversion results in a high-density cluster of protein conformations inside the basin. So, to get an initial estimate of the states, we treat the input dataset D as a set of conformations and use the K-medoids algorithm to partition the conformations in D into K clusters, where K is a pre-specified parameter. K-medoids forms compact clusters by minimizing the sum of intra-cluster distances between conformations [4] under the same distance d used for specifying the emission probability distributions (7). The center of a cluster C is a conformation $q \in C$ that minimizes the sum of distances from q to other conformations in C.

Each cluster then becomes a state of our initial model Θ_0. Using the cluster labels of the conformations in D, we can easily compute the prior probabilities Π and transition probabilities A for Θ_0 by simply counting. To get the emission probability $e_i(q) = N(q|\mu_i, \sigma_i^2)$, we set μ_i to the center of the cluster corresponding to state s_i and σ_i^2 to the variance of conformations in this cluster.

Optimization. We use Θ_0 to initialize the EM algorithm and search for a K-state HMM Θ that maximizes the likelihood $p(D|\Theta)$. EM iterates over two steps, expectation and maximization, to improve the current model until no further improvement can be achieved. Inspection of (1) shows that our main difficulty is the summation of all possible state assignments to the conformations $(q_0, q_1, \ldots, q_T)$ along a trajectory D_i. Performing this summation by brute force takes time $O(K^T)$, which is exponential in the length T of the trajectory. EM overcomes this difficulty through dynamic programming. See [4] for details.

The number of states. The number of states K controls the model complexity. It must be specified in both K-medoids clustering and EM. A complex model with many states can in principle fit the data better, thus achieving higher likelihood. However, it may suffer from overfitting when there is insufficient data. A complex model is also more difficult to analyze and understand. Typically a simple model is preferred when it does not sacrifice much predictive power. To choose a suitable K value, we pick a small random subset D' of D as a test dataset. We start with a small K value and gradually increases it until the likelihood $p(D'|\Theta)$ levels off. It is important to note that we can perform such a search over model complexity because our likelihood criterion enables us to compare models with different number of states.

6 RESULTS

We tested our approach on 2-D synthetic energy landscapes and on two extensively studied proteins, alanine dipeptide and villin. The results are reported in the subsections below.

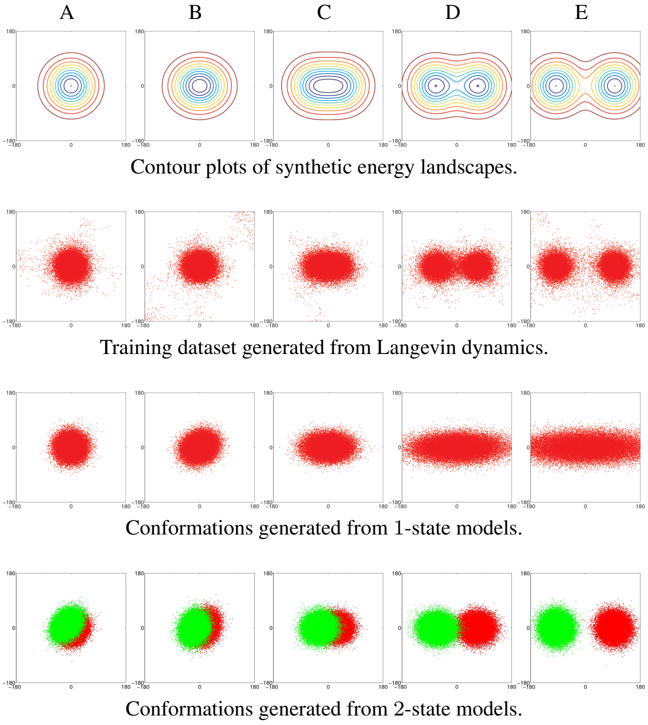
6.1 Synthetic Energy Landscapes

Synthetic energy landscapes are useful for testing our algorithms in controlled settings where the ground truth is known. In particular, we want to examine whether our likelihood criterion and model construction algorithm can identify simple models with strong predictive power.

We created a series of five energy landscapes in two dimensions (Fig. 1). Landscapes A and B each contains a single energy basin, but B's basin is slighted more elongated. Landscapes C, D, and E each contains two basins, but the separation between the basins varies. For each landscape, we used Langevin dynamics to generate 1000 trajectories of 200 time steps each. We set aside half of the trajectories as the training dataset for model construction and the other half as the test dataset D' for checking the quality of the model constructed.

For each landscape, we built models with increasing number of states. In all the models, the resolution h is 10 simulation time steps. The distance measure d used in defining the emission probabilities is the Euclidean distance in the plane.

Fig. 2 plots the scores of all the models. The score is the average log-likelihood of a model for a single transition step along a trajectory. It is computed by dividing the log-likelihood of a model given D' by the total number of conformations in D'. Fig. 2 shows that for landscape A, which contains only 1 basin, the 1-state model is slightly better than the 2-state model. As we move from landscape A to E, the predictive power of the 1-state model degrades. The 2-state model performs fairly well on all five energy landscapes. Fig. 1 shows the differences between the 1-state and 2-state models by simulating them and plotting the resulting conformations. Fig. 2 also shows that increasing the number of state beyond 2 has negligible benefit. This is expected, because the underlying energy landscapes have at most two basins.



Contour plots of synthetic energy landscapes.

Training dataset generated from Langevin dynamics.

Conformations generated from 1-state models.

Conformations generated from 2-state models.

Fig. 1. Five synthetic energy landscapes and the corresponding models.

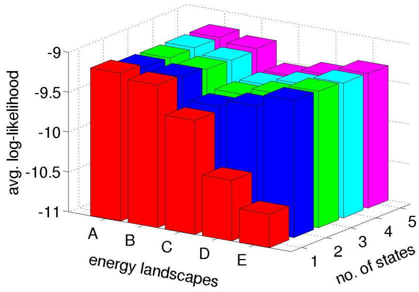


Fig. 2. Average log-likelihood scores of the models for synthetic energy landscapes.

6.2 Alanine Dipeptide

Alanine dipeptide (Ace-Ala-Nme) is a small molecule widely used for studying biomolecular motion, as it is simple and exhibits an the extensive range of torsional angles. We use the same dataset as that from a previous study [7]. It consists of 1000 MD simulation trajectories totaling 20 ns in length. Again, we divide them equally into training and test datasets.

We built models with with up to 7 states. They are named A1 to A7. As alanine dipeptide is very small, its motion is fast. So the time resolution h of the models is set to 1.0 ps. A conformation of alanine dipeptide is specified by three backbone torsional angles (ϕ, ψ, ω), and the distance between two conformations is defined as the root squared angular differences between the corresponding torsional angles.

The conformation space of alanine dipeptide can be manually decomposed into 6 disjoint region, each corresponding to a metastable state. This well-accepted decomposition has led to several dynamic models of alanine dipeptide [7, 8]. For comparison, we built a 6-state model based on the same manual decomposition. During the model construction, instead of applying K-medoids, we group conformations into a cluster if they belong to the same disjoint region of the manual decomposition. Other steps of the construction algorithm remain the same. The resulting model is named M6.

Fig. 3 plots the average log-likelihood scores of all the models constructed. Models A3 to A7 all achieve scores comparable to that of M6. The interesting finding is that although the score jumps substantially as we move from A1 to A3, the score of A3 is almost as good as those of A6 and M6. This indicates, in particular, that for predicting the motion of alanine dipeptide, the simpler 3-state model A3 is almost as good as the 6-state model M6, which is obtained from the well-accepted manual decomposition of the alanine dipeptide conformation space!

To see the differences between A3 and M6, we simulated the two models and plotted the resulting conformations (Fig. 4). Both models capture accurately the frequently visited regions of the

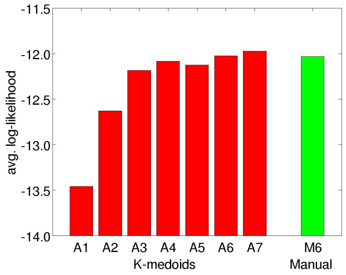


Fig. 3. Average log-likelihood scores of alanine-dipeptide models.

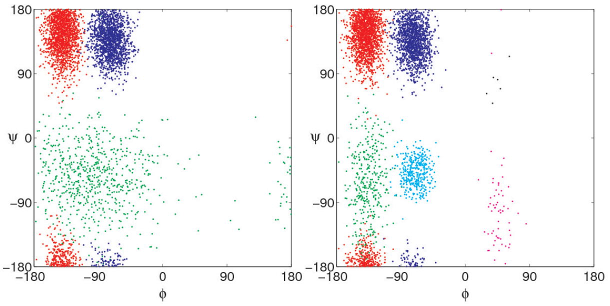


Fig. 4. Conformations generated from the 3-state model A3 (left) and the 6-state model M6 (right).

Table 1. Estimated MFPTs between α_R and $\beta/C5$ regions of the alanine dipeptide conformation space.

	MFPT (ps)	
	A3	M6
$\alpha_R \to \beta/C5$	26.5	28.5
$\beta/C5 \to \alpha_R$	187.0	124.0

conformation space, shown in red and blue in Fig. 4. These densely sampled regions correspond to energy basins that dominate the long-term dynamics, and accurate modeling of these regions is crucial. Next, for A3, the conformations shown in green capture a large, but less frequented region of the conformation space. Although M6 models the same region as two closely spaced clusters of conformations, the overall density and the location of the conformations are similar in both models. Finally, M6 also models the rarely visited region between $0 < \phi < 90$. Due to the transient nature of the protein in these conformations, any contribution from this additional complexity is minimal in terms of any long-term dynamical phenomena that are experimentally measurable. Therefore, the leveling of the log-likelihood scores with respect to the number of states corresponds to the diminishing returns in modeling any additional complexity of the long-term dynamics.

To further validate our models, we used both A3 and M6 to compute MFPTs between the α_R and $\beta/C5$ regions of the conformation space. We designate conformations with ($\phi = -70 \pm 15, \psi = -40 \pm 15, \omega = 180 \pm 15$) to be within the α_R region, and conformations with ($\phi = -140 \pm 15, \psi = 160 \pm 15, \omega = 180 \pm 15$) to be within the $\beta/C5$ region. Although the results for A3 and M6 differ somewhat in details, they are consistent (Table 1). Both indicate that the transition from α_R to $\beta/C5$ is roughly an order of magnitude faster than the reverse transition. This matches well with the results reported by Chekmarev *et al.* [5].

To assess the efficiency of our algorithm for MFPT computation (Section 4), we also computed the MFPTs by explicitly generating simulation trajectories from our constructed models. It took our algorithm less than 1 second to compute one MFPT, as the alanine dipeptide models are all very simple. In comparison, it took 120 seconds to generate a sufficiently large number of simulation trajectories from the same HMM in order to bring the standard deviation of the MFPT estimate down to 1% of its value.

6.3 Villin

Here we present results on data collected by the Folding@home project on the fast-folding variant of the villin headpiece HP-35 NleNle. This data consists of 410 MD trajectories initiated from 9 unfolded conformations denoted by I_k, $k = 0, ..., 8$. We set aside half of the trajectories for training and the other half for testing.

Due to the higher dimensionality of the conformation space (35 residues) and the huge number of conformations, we define a graph-based distance d between conformations (recall that d is used by the clustering algorithm and the emission probabilities) that better captures the kinetics than the usual RMSD metric and avoids computing all pair-wise distances between conformations. In order to define this distance, we first group conformations into microstates. We sample 8000 conformations uniformly along the trajectories in the training dataset to serve as the microstate centers. The rest of the conformations are then clustered to the nearest microstate based on the RMSD of all heavy atoms to the microstate centers. As in [22], we approximate the kinetics in high-dimensional space by assuming that the protein can only transit between microstates that are close RMSD wise. This leads us to create a graph where each microstate is connected to each of its nearest neighbors with an edge of length equal to the RMSD between the two microstate centers. Only a small number of nearest neighbors (less than 10) needs to be connected to ensure that the graph is connected. The distance between two microstates is then defined as the shortest-path distance between them in the graph, and the distance between two conformations is the distance between the microstates they belong to.

We constructed HMMs of different number of states, all at $h = 5$ ns. The average log-likelihood scores (Fig. 5) improve significantly when the number of states grows from 1 to 20. They only improve gradually between 20 and 200 states. Beyond 200 states, the scores remain approximately constant.

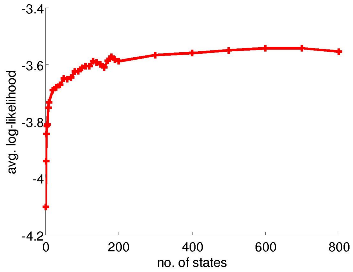
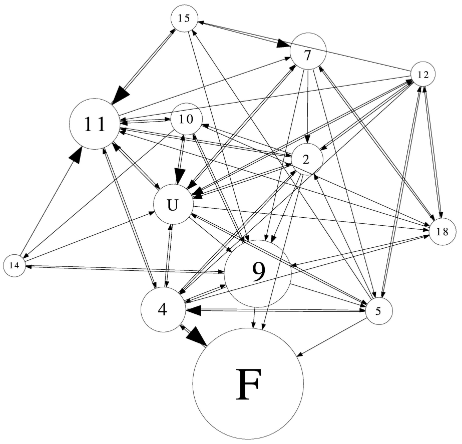


Fig. 5. Average log-likelihood scores for villin HMMs at $h = 5ns$.



a) State transition diagram

b) Most probable folding transitions

State Sequence	Probability
$U \rightarrow 4 \rightarrow F$	0.0281
$U \rightarrow 9 \rightarrow F$	0.0143
$U \rightarrow 5 \rightarrow 4 \rightarrow F$	0.00702
$U \rightarrow 9 \rightarrow 4 \rightarrow F$	0.00677
$U \rightarrow 11 \rightarrow 4 \rightarrow F$	0.00436

Fig. 6. Main state transitions of the 20-state model at $h = 5$ ns. The size of each node is proportional to the stationary probability of the corresponding state. The size of each arrow is proportional to the transition probability. States with stationary probability less than 0.01, self transitions, and transitions with probabilities less than 0.0005 are not shown to avoid clutter the diagram. Given an initial conformation, q_U, state U is the most likely state ($p(U|q_U) = 0.54$ on average). Given the native conformation, q_F, under stationary distribution, state F is the most likely state ($p(F|q_F) = 0.49$).

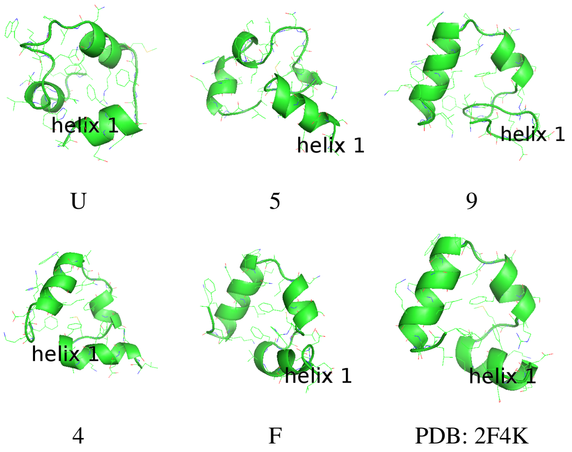


Fig. 7. Conformations of state centers in Fig. 6. Helix 1 in U is in the wrong orientation despite having a high degree of helical structure.

Table 2. MFPT for each of the nine initial conformations in the MD dataset. Each conformation I_k is used to initiate a set of trajectories.

a) Slow folding initial conformations

MFPT (μs)						
I_0	I_1	I_2	I_3	I_5	I_6	I_8
8.99	14.2	10.9	10.7	8.75	10.5	10.8

b) Fast folding initial conformations

MFPT (μs)	
I_4	I_7
4.87	4.12

To see the main features of villin dynamics, consider the the 20-state model. The model reveals that given the initial conformations, state U in Fig. 6*a* is the most likely state of the protein and helix 1 is likely to be in the wrong orientation despite having attained a significant degree of helical structure (see Fig. 7). Once in state U, several transitions to other states are possible with similar probabilities. The common feature among these states is that both helix 2 and helix 3 have achieved a high degree of helical structure (61% for helix 2 and 78% for helix 3 achieved on average), whereas helix 1 is totally disordered in half of the states. Therefore, attaining *both* the helical structure and the correct orientation for helix 1 appears to be the barrier limiting folding. In [11], the presence of helix 1 in the initial conformations I_4 and I_7 was considered one of the possible reasons why trajectories starting at these conformations fold faster (on average) than trajectories starting at the other initial conformations.

The folding of villin has been extensively studied in the wet lab using various experimental techniques. In order to obtain a reliable estimate of the experimental quantities, we define the set of folded conformations to be the 5% microstates that are closest to the experimentally determined native structure (PDB: 2F4K), and compute the MFPT for the set of initial conformations. The emission probability distribution of the folded set is then given by: $e_i(F) = \frac{\sum_{q_i \in F} p(q_i|s_i)}{\sum_{q_i \in G} p(q_i|s_i)}$, where F is the set of folded microstates and G the set of all microstates.

The computed MFPT for the initial conformations I_0 through I_8 in Table 2 are in the same microsecond range of the experimental $4.3\ \mu$s measured using laser temperature jump by Kubelka *et al.* [17], and the $10\ \mu$s measured using NMR line-shape analysis by Wang *et al.* [28]. In addition, the MFPTs for I_4 and I_7 are smaller, which is consistent with the computational analysis of Ensign *et al.* in [11]. We also attempted to compute each MFPT by generating trajectories with the HMM and averaging the trajectory lengths. However, instead of the less than 1 minute it takes to solve (3), after 30 minutes of generating trajectories, the estimated MFPT is still two orders of magnitude less than the values in Table 2.

7 CONCLUSION

During the past decade, there has been increasing interest in graphical models of protein motion at long timescales. Most recently, the focus has been on cell-based Markov dynamic models built from pre-computed MD simulation data. Existing methods, however, suffer from two main shortcomings. First, defining states by partitioning the conformation space of a protein into disjoint cells results in violation of the Markovian property. Second, there is no systematic criterion for evaluating the quality of a Markov dynamic model. This paper addresses these two shortcomings by defining states as probabilistic distributions of conformations. This reflects the view that a single conformation does not contain enough information to be assigned to a unique state. The resulting HMM-based modeling framework evaluates model quality by the likelihood of a model given a test dataset of simulation trajectories. This allows for comparison of models, in particular, with different number of states. Test results on two widely studied proteins validate our approach.

One important issue remaining is to scale up our approach to handle larger proteins. MD simulation is computationally expensive, but advances in computer technology are making it more affordable than before, and large simulation data repositories will become readily available over time. Increasingly, the future challenge will be to gain biological insights from this data by building simple and yet powerful models. A computational bottleneck in our current model construction algorithm is distance computation during the clustering. However, we believe that the graph-based method developed for villin is reasonably scalable, as it avoids performing all pair-wise distance computations.

As we scale up to larger proteins, the dynamics of protein motion will also become more complex. For large proteins, it is likely that motions at different timescales contribute to different biological functions. A hierarchy of HMMs constructed at different timescales may capture such multi-timescale dynamics.

ACKNOWLEDGEMENT

Jean-Claude Latombe's research is partially supported by two grants from the Academic Excellence Alliance program between King Abdullah University of Science and Technology(KAUST) and Stanford University. The authors thank Vijay Pande and Nina Singhal who provided us MD simulation data for alanine dipeptide and villin.

REFERENCES

[1] A. Amadei, A.B. Linssen, and H.J. Berendsen. Essential dynamics of proteins. *Proteins: Structure, Function, and Genetics*, 17:412–425, 1993.

[2] N.M. Amato, K. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *J. Comput. Biol.*, 10:239–255, 2003.

[3] M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, J.C. Latombe, and C. Varma. Stochastic roadmap simulation: an efficient representation and algorithm for analyzing molecular motion. *J. Comput. Biol.*, 10:257–281, 2003.

[4] G. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[5] D.S. Chekmarev, T. Ishida, and R.M. Levy. Long-time conformational transitions of alanine dipeptide in aqueous solution: continuous and discrete-state kinetic models. *J. Phys. Chem. B*, 108:19487–19495, 2004.

[6] T.H. Chiang, M.S. Apaydin, D.L. Brutlag, D. Hsu, and J.C. Latombe. Predicting experimental quantities in protein folding kinetics using stochastic roadmap simulation. In *Proc. ACM Int. Conf. on Research in Computational Molecular Biology (RECOMB)*, 2006.

[7] J. Chodera, N. Singhal, V. S. Pande, K. Dill, and W. Swope. Automatic discovery of metastable states for the construction of markov models of macromolecular conformational dynamics. *J. Chem. Phys.*, 126:155101, 2007.

[8] J. Chodera, W. Swope, J. Pitera, and K. Dill. Long-time protein folding dynamics from short-time molecular dynamics simulations. *Multiscale Modeling & Simulation*, 5:1214–1226, 2006.

[9] R. Du, V. Pande, A.Y. Grosberg, T. Tanaka, and E. Shakhnovich. On the transition coordinate for protein folding. *J. Chem. Phys.*, 108:334–350, 1998.

[10] R. Elber. Long-timescale simulation methods. *Curr. Opin. Struct. Bio.*, 15:151–156, 2005.

[11] D.L. Ensign, P.M. Kasson, and V.S. Pande. Heterogeneity even at the speed limit of folding: large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. *J. Mol. Biol.*, 374:806–816, 2007.

[12] D.E. Shaw et al. Anton, a special-purpose machine for molecular dynamics simulation. In *Proc. Int. Symp. on Computer Architecture*, 2007.

[13] V.S. Pande et al. Atomistic protein folding simulations on the hundreds of microsecond timescale using worldwide distributed computing. *Biopolymers*, 68:91–109, 2002.

[14] T. Haliloglu, I. Bahar, and B. Erman. Gaussian dynamics of folded proteins. *Phys. Rev. Lett.*, 79:3090–3093, 1997.

[15] X. Huang, Y. Yao, G.R. Bowman, J. Sun, L.J. Guibas, G. Carlsson, and V. Pande. Constructing multi-resolution markov state models (MSMs) to elucidate rna hairpin folding mechanisms. In *Proc. Pacific Symp. on Biocomputing*, 2010.

[16] L.E. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics & Automation*, 12:66–580, 1996.

[17] J. Kubelka, W. A. Eaton, and J. Hofrichter. Experimental tests of villin subdomain folding simulations. *J. Mol. Biol.*, 329:625–630, 2003.

[18] A.R. Leach. *Molecular Modeling: Principles and Applications*. Prentice Hall, 2001.

[19] M. Levitt, C. Sander, and P.S. Stern. Protein normal-mode dynamics: Trypsin inhibitor, crambin, ribonuclease and lysozyme. *J. Mol. Biol.*, 181:423–447, 1985.

[20] G. Lois, J. Blawzdziewicz, and C. O'Hern. The free energy reaction path theory of reliable protein folding. *Biophys. J.*, 96:589a–590a, 2009.

[21] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 3rd edition, 2009.

[22] E. Plaku and L. E. Kavraki. Nonlinear dimensionality reduction using approximate nearest neighbors. In *SIAM Inter. Conf. on Data Mining*, pages 180–191, 2007.

[23] J.-E. Shea and C.L. Brooks III. From folding theories to folding proteins: A review and assessment of simulation studies of protein folding and unfolding. *Annu. Rev. Phy. Chem*, 52:499–535, 2001.

[24] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexible ligand binding. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, pages 252–261, 1999.

[25] N. Singhal, C.D. Snow, and V.S. Pande. Using path sampling to build better markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *J. Chem. Phys.*, 121:415–425, 2004.

[26] H. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, New York, 1994.

[27] M. Teodoro, G.N. Jr. Phillips, and L.E. Kavraki. A dimensionality reduction approach to modeling protein flexibility. In *Proc. ACM Int. Conf. on Computational Molecular Biology (RECOMB)*, pages 299–308, 2002.

[28] M. Wang, Y. Tang, S. Sato, L. Vugmeyster, C. J. McKnight, and D. P. Raleigh. Dynamic NMR line-shape analysis demonstrates that the villin headpiece subdomain folds on the microsecond time scale. *J. Am. Chem. Soc.*, 125(20):6032–6033, 2003.